# Modeling the Extended Enterprise: A Comparison of Different Modeling Approaches

Orsolya Szegheo
Ph.D. Student
Norwegian University of Science and Technology
Department of Production and Quality Engineering
N-7491 TRONDHEIM, NORWAY
Orsolya@ipk.ntnu.no

Bjørn Andersen, Ph.D.
Associate Professor
Norwegian University of Science and Technology
Department of Production and Quality Engineering
N-7491 TRONDHEIM, NORWAY
Bjorn.Andersen@ipk.ntnu.no

ABSTRACT: The industrial development during the 1980's and 1990's has changed the market conditions for industrial activities. The firms are blurring their traditional boundaries and entering into close collaboration with other firms. The literature refers to the new way of collaboration as the extended enterprise. This paper compares different enterprise modeling approaches, such as process modeling (ARIS), active knowledge modeling, object-oriented and agent based modeling. The study aims to explore whether these approaches are capable of modeling the extended enterprise, i.e. whether they have the capabilities to model a network of enterprises and not only one individual enterprise.

KEYWORDS: extended enterprise, enterprise modeling, process modeling, object oriented modeling, agent based modeling, active knowledge modeling

## 1 INTRODUCTION

Strategic alliances, networks and other form of quasi-integration are the words that are depicting most adequately the changed forms of industrial activities. The term extended enterprise was invented to describe the new way of coordination and collaboration. According to Browne, the extended enterprise is a form of closer coordination in the design, development, costing, and the coordination of respective manufacturing schedules of cooperating independent manufacturing enterprises. The extended enterprise relies heavily on the extensive use of technical standards, advanced computer communication facilities, and Electronic Data Interchange (EDI) techniques (Browne, 1998), see Figure 1.

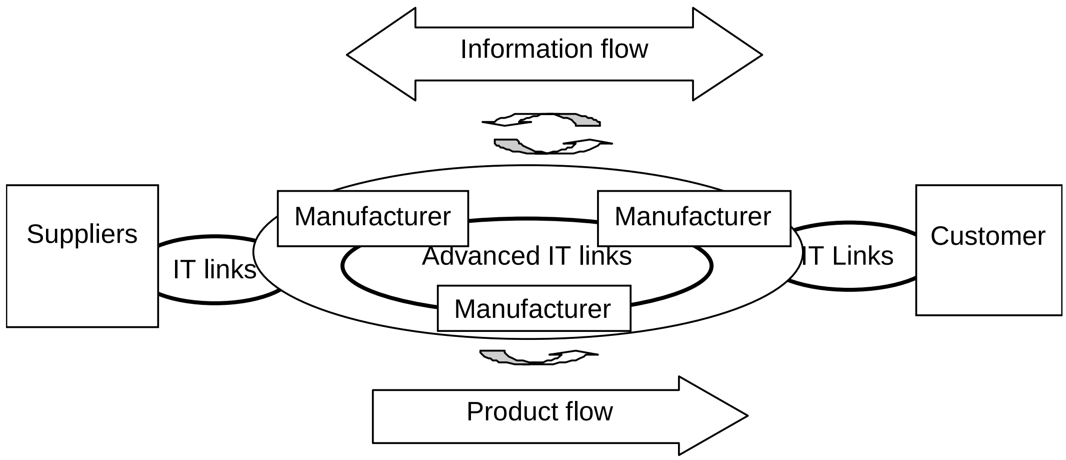


**Figure 1** The Extended Enterprise

This paper analyses how the originally single enterprise focused enterprise modeling can be used for the extended enterprise. Enterprise modeling aims at making people understand, communicate, develop and cultivate solutions to business problems. In basic terms, the main objective of modeling is to make a simplification of the real world situation through abstraction. A good model must display the same characteristics or properties as the slice of the world from which it has been extracted. However, because a model is much simpler, it can more easily be studied and manipulated to yield a solution (Gigh, 1991).

Taking into account the experience of enterprise modeling we hope that modeling the extended enterprise enables the understanding of the activities performed in the enterprise network and particularly helps to understand and improve the relationships that connect the enterprises. Though the extended enterprise is a quite new phenomenon, business practitioners have already started to explore how it can be improved. The inspiration for improvement might come from better understanding of the partner’s operations and a clear understanding of the activities taking place to manage the relationship between the companies.

In this paper, various approaches will be studied and analyzed as to whether they have the capabilities for modeling the extended enterprise.

# 2 MODELING APPROACHES

## 2.1 Active Knowledge Modeling

In the “Old World of Business” almost everything was predictable. It was assumed that with “best practices”, the organization could retain its effectiveness. Meanwhile, the “New World of Business” has arrived and our business environment is characterized by uncertainty and an unpredictable future. The participants need the capability to solve tasks in a changing environment. The focus is on “*doing the right things*” rather than on “*doing things right*.” Defining the problem itself is as important as finding the right answer.

Knowledge management caters to the critical issues of organizational adaptation, survival and competence in the face of increasingly discontinuous environmental change. Essentially, it embodies organizational processes that seek synergistic combination of data and information processing capacity of information technologies, and the creative and innovative capacity of human beings (Malhotra, 1998).

The trend in business is to substitute human activities with machines. With the programmed systems, the organization aims to optimize and make its activities more effective. The problem with such a system is that it does not yet have the capabilities to understand the environment's complexity the way a human being can. Knowledge management focuses on how new technology and human knowledge can be combined.

A knowledge model is the human's knowledge of a subject transferred into a model of some sort. An Active Knowledge Model (AKM) implies that the knowledge changes dynamically over time (Prytz, 1995). An Active Knowledge Model (AKM) is a reflection of common knowledge and understanding of processes and supporting aspects of organization, product, workflow, and system structures (Lillehagen, 1995).

The AKM enables the user to extend and reengineer meta-data, domain languages, and views. The modeling can start with simple views and then, as more knowledge is accumulated, the model can be extended and more detail can be added. Figure 2 shows the steps in the computer-supported human learning cycle of individuals, and how the learning is supported by AKMs. It is only Data and Information that is stored in the AKM. The other steps; Knowledge through to Methods are achieved by humans by the use of AKMs.

The competence of the enterprise is enriched over time, as the enterprise is able to capture more and more of its corporate knowledge and intellectual capital. By creating knowledge models and improving the existing ones, one enhances the knowledge of the enterprise. Knowledge modeling enables knowledge cultivation and reuse.

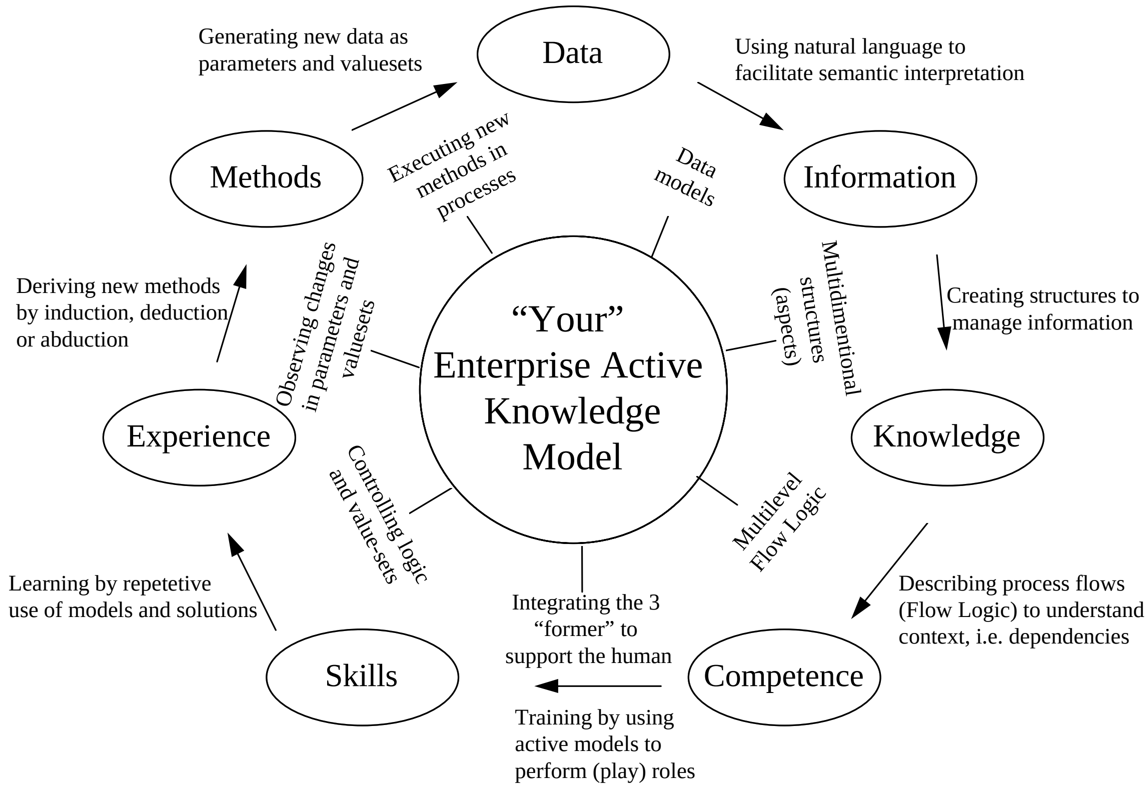


**Figure 2** Computer supported human learning

## 2.2 Process modeling

A business process is a procedure designed to add value in an organization. A business process consists of two function classes. The first one is the transformation of input data into output data. This is a typical office activity. The other one is the transformation of input material into output material. This process is called manufacturing.

In process based modeling, processes in the domain of interest are modeled. The whole process is included; flow of control, flow of materials, and flow of information, without any consideration of organizational boundaries.

There are several methodologies (CIMOSA, PERA, ARIS, etc.) and tools that facilitate business process management. As an example in this paper, ARIS (Architecture of Information Systems) will be studied in more detail. ARIS has been developed by Professor Scheer at the University of Saarbrücken (Scheer, 1992). The overall ARIS structure is very similar to CIMOSA, but instead of focusing on computer-integrated manufacturing systems, it deals with more traditional business-oriented issues of enterprises (such as order processing, production planning and control, inventory control, etc.). The focus is essentially on software engineering and organizational aspects of integrated enterprise system design.

Processes, activities, events, conditions, and organizational units are included in the description of a business process. In order to reduce complexity, ARIS is divided into views that represent

discrete design aspects and can be handled independently. Events such as an “order” are information objects that are represented by data, reference field conditions such as “customer status” are also represented by data. Conditions and events thus form the data view. The functions to be performed and their relationships form the function view. The structure and relationships between staff members and organizational units constitute the organization view. The control view is used to represent the relationships between the components.

### 2.2.1 Modeling the Extended Enterprise with ARIS

Modeling the enterprise network could facilitate the creation of enhanced understanding of the business processes of the extended enterprise and relationships that extend across the boundaries of the enterprises.

In the extended enterprise different business processes from the area of design, development, costing, scheduling, etc. are closely coordinated between independent manufacturing enterprises. The main principle in modeling the processes in the extended enterprise is that the value–adding process across the enterprises should be taken as a whole. This should avoid the sub-optimization of one part (or partner) at the expense of the other. It makes the modeling more difficult because the model builder has to cross the boundaries of the enterprises. The information collection becomes more difficult and the validity of the information is not guaranteed. When the information source is not located inside the enterprise, the information quality is doubtful because the number of uncertainty factors increases.

If the enterprises cooperating in the extended enterprise decide to reengineer some of the business processes of the extended enterprise, they will face the problem that in the extended enterprise, a process consists of sub-processes that are “owned” by different individual enterprises. Using a process-modeling tool such as ARIS, they will have to overcome the problem that the various parts of ARIS, such as functional, organizational, data, and control elements, are distributed in the extended enterprise. They are not found inside the boundaries of one enterprise, but are located in different places. The modeling should take place in a cooperative way across multiple locations. It requires special techniques for modeling, common ontology, and global accessibility.

## 2.3 Object oriented modeling

Model building is a time consuming and costly activity. The model builder has to strive for a model that is robust enough to stand against change and can keep its validity over a longer period of time. The question is how to achieve this if one does not know what kinds of changes are going to happen and which elements will be changed. One approach could be the separation of the building elements and their occurrence. Treating the elements as separate objects enables easier model maintenance and model reconfiguration. As Hill states, experience show that data handled by the software is more stable than the process associated with it (Hill, 1996).

The object is an abstraction of something in a problem domain, reflecting the capabilities of a system to maintain information about it, interact with it, or both, an encapsulation of Attribute values and their exclusive Services. The object oriented approach gives priority to the data

handled by the software, and tackles modeling by identifying a collection of objects of the problem domain. Each object has a specific task and the problem is solved by object interaction.

The object oriented approach is defined by Coad and Yourdon (1991) in the following way:

Object-oriented = Classes and Objects
+ Inheritance
+ Communication with messages

The class is a description of one or more Objects with a uniform set of Attributes and Services, including a description of how to create new Objects in the Class. Inheritance makes use of the principles of generalization and specialization, it allows a subclass to have attributes and methods previously defined in a super-class representing a generalization of the subclass (itself considered as a specialization of the super-class) (Hill, 1996).

Object and classes are relatively stable while Attributes and Services could change rapidly. In an object oriented model if one of the objects is changed it is enough to make the adjustment only once because each object is defined only once. Once an object is described, the needed information is available for various retrospective inquiries. If changes need to be made to an object, changes are made in the class library.

Each object has a state and behavior. The possible states and behavior of the objects are defined by the classes. Each object is an instance of a class. All the instances of one and the same class have the same characteristics and the same reactions, but each instance is identified by the value of its attributes.

Message sending is the only communication channel between the objects. A message constitutes a request to activate one of the methods of an object. The message must correspond to a part of the behavior of the object, otherwise an error occurs (Hill, 1996).

Encapsulation and late binding are fundamental principles of object-orientation. Encapsulation is the process of hiding all the details of an object that do not contribute to its essential characteristics (Booch, 1991). The object can be accessed only by the offered operations and methods. Late binding refers to the ability of delaying the process of relating data and the procedure to be applied until execution time (Vernadat, 1996).

### 2.3.1 Modeling the extended enterprise by object oriented modeling

The main difference between object oriented analysis (OOA) and process oriented modeling (POA) is that in OOA all characteristics of a certain object are identified, while in POA the model builder identifies only those characteristics that play a certain role in the given situation. In the extended enterprise one resource or element could be used many times by different actors and in different processes. Defining an object library for the extended enterprise will enable reusability and speed up the model building.

The business environment is not static, it is a subject of continuous change, and the model has to be up-dated continuously to avoid obsolescence. When all the objects constituting the model are defined only once, in one library, the model up dating and model altering becomes easier.

## 2.4 Agent based system

The Multi Agent System (MAS) is constituted by loosely coupled agents. Each agent has a goal, can perform actions, can control resources, and has a strategy in order to achieve some goals. The agents communicate with each other, sense the environment, and act upon it. They try to attain local goals in complex, dynamic environments. Searching for a solution is done by interchanging messages according to a negotiation protocol.

Cetnarowicz (1997) defines the MAS agent functioning the following way:

- The agent observes the environment and builds a model of it in its mind: m = I (M, V). This leads to creation of the model m of the environment V, I is the observation function.
- The agent selects a strategy that gives an optimal solution: s = F (M, Q, S, V). This gives a selection of an optimal strategy. M, Q, and S are sets of models, goals, and strategies, V is the environment, and F is the decision function enabling a selection of an optimal strategy.
- The agent executes the selected strategy: V'= X (s, V). This is a realization of an optimal strategy s in the environment V, the X function represents operation of the realization of an agent's strategies.

The MAS system design can start at a very high level and can be elaborated through detailed abstraction levels. The top level gives a general picture of the analyzed system. Through abstraction, we can create a more complicated system, adding more properties to the system. The lowest abstraction level, which defines the final version of the system, may be specified in programming languages. Irrespective of the abstraction level, the following have to be defined to design a MAS (Cetnarowicz, et. al. 1997):

- Living space for the agents.
- Agents of several types (kinds).
- Relations between agents and the living space and among agents themselves.

An important feature of the decentralized systems is that the elements of these systems have individual goals. The degree to which the individual goals have been attained is usually subject to an evaluation of some kind, and this evaluation is the basis for choosing the agent's decision that are optimal in a given situation.

The way in which the individual agents affect the values of the other agents' criteria or the value of an objective function describing the system taken as a whole, has been adopted as a basis for further division of the terms of cooperation into classes. (Nawarecki et. al., 1997) The classification is showed in Figure 3.

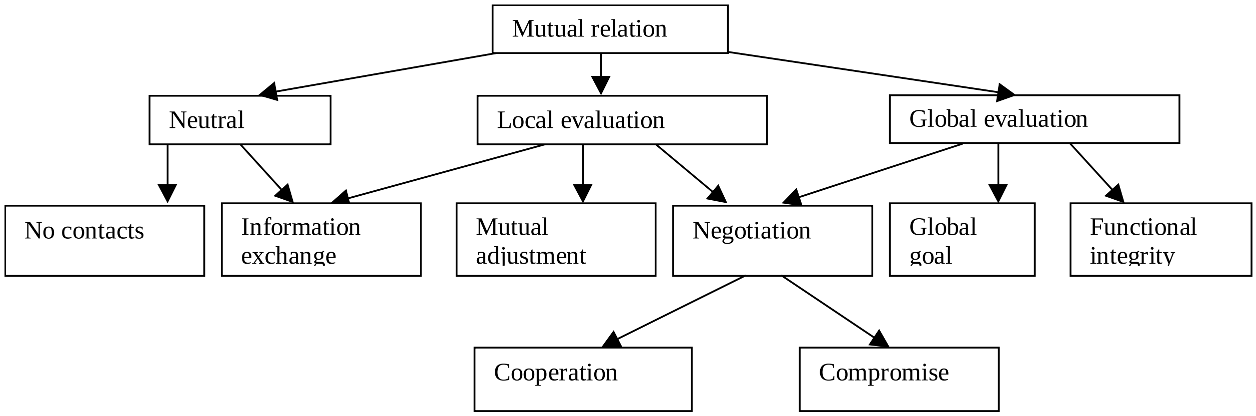


**Figure 3** Division of cooperation into classes

The literature distinguishes between different agent types, among others reactive, adaptive, and cognitive agent types are mentioned (Kjenstad, 1998, Bittencourt, 1997). Reactive agents are typically simple and in general, their problem solving capabilities are limited to a functional deterministic behavior. In a reactive MAS, the intelligent behavior is not a property of any agent in particular, it is an emergent characteristic of the whole community. An adaptive agent is able to improve over time, the agent becomes better at achieving its goals with experience, and able to adjust to a changing environment, such as by dynamically recognizing and making use of new actors in the agent community. Cognitive agents are typically complex and able to present some intelligent behavior by themselves, independently of the agent community. They usually dispose of a more or less complex internal environment model and interact using structured communication protocols, such as the Contract Net Protocol.

### 2.4.1 Modeling the extended enterprise by MAS

Any kind of network could be described by nodes and relationships. Intelligent, communicating agents govern the nodes and relationships. The extended enterprise could be seen in a way where each enterprise in the network is an independent element which has an existence on its own and is able to cooperate with other elements. Each enterprise in the network is represented by an agent that has an individual goal and its own strategy. The user could define different scenarios and run tests, trying out what happens in the network if one of the agents changes its strategy.

Another way of representing the extended enterprise with a MAS system could be the following. Each agent of the MAS performs one or more supply chain functions, and each coordinates its actions with other agents. The MAS of the enterprise network could be defined trough abstraction. The main abstraction levels could be the strategic, the tactical, and the operational level of the management system. The strategic level defines the enterprise network, i.e., the selection of suppliers, transportation routes, core competencies, manufacturing facilities, production levels, warehouses, etc. The tactical level plans and schedules the enterprise network to meet actual demand. The operational level executes plans.

The following extended enterprise framework deals mainly with the strategic level and does not define low level agents on the tactical and operational level. On the strategic level, we could

distinguish between three main kinds of agents; agents who take care of the activities of one individual enterprise, called functional agents; agents who take care of the whole network, called network agents; and agents who take care of the relationships, called relationship agents.

*Functional agents:*

- Order acquisition agent, connection between the customer and the enterprise, has contact with the network agent.
- Scheduling agent.
- Resource agent.
- Dispatching agent.
- Transportation agent.
- Information agent, keeps functional agents aware of any events and changes, distribute information.

*Network agents:*

- Core competency agent, gathers information about the core competencies available in the network.
- Environmental agent.
- Logistics agent.
- Contract agent, securing the order fulfillment.

*Relationship agents:*

- Legal issues and conflict resolving agent.
- Communication, technical standards, procedures, and infrastructures agent.

# 3 CONCLUSION

Before choosing the modeling approach, the model builder or the modeling team has to define precisely what is the purpose of the model. Each modeling technique was developed to satisfy some particular needs. The model builder and the user of the model can benefit most from the model if the right tool has been chosen for their purpose. Each tool has its application area where it can be used to its best advantage. In this paper, different modeling approaches are analyzed and suggestions are made as to how these approaches can be used for modeling the extended enterprise.

Process modeling can be used when the emphasis is on improving the quality of relationships in the extended enterprise. The modeling activity should be a mutual undertaking, all the partners involved in the relationship have to take part in mapping the process. That way, hopefully the results of the modeling activity will be accepted mutually and none of the partners will feel that the solution was forced by the counterpart. The modeling requires special expertise because the modeling takes place in a locally distributed environment between people with different cultural backgrounds. In that context, cultural background refers to the fact that each enterprise has its

own professional terminology that is not necessarily interpreted the same way by others who are not members of the particular organization. The mutual modeling activity requires a common ontology that is interpreted by all participants the same way, modeling techniques and tools that supports collaborative modeling, and a tool or tools that are accessible and accepted by all the participants.

The object-oriented approach accelerates model building. The objects of the extended enterprise domain can, once they have been defined, be reused in many different scenarios. Though at first it takes long time to define all the attributes of an object, later when the object is reused, valuable time can be saved by having a predefined object library with all the needed objects that constitute the system.

The extended enterprise is a network of cooperating partners, this description corresponds to the description of a Multi Agent System (MAS). The MAS consists of cooperating agents. For some modeling scenarios, MAS is an applicable approach. For example one could experiment how different enterprises' strategies influence the performance of the network. But if there is close collaboration between the participants in the extended enterprise, each enterprise has its own strategy. Identifying the different strategies of the enterprises, it can be measured how altering one of the participants' strategy would influence the performance of the extended enterprise.

Though the above listed approaches are fundamentally different from each other, there are some universal modeling principles that must be followed in any case:

- Every member of the modeling team must feel comfortable with the chosen modeling approach.
- Careful planning has to precede any kind of modeling activity.
- Information collection is an essential part of the modeling because the model validity depends on the quality of the information.

**Reference list**

Barbuceanu M, Fox M S.: The Architecture of an agent based infrastructure for agile manufacturing, Enterprise Integration Laboratory.

Bittencourt G, Augusto Cesar Pinto Loureiro da Costa: Expert –COOP: An environment for cognitive multi-agent systems. 1997, IFAC/IFIP Conference on Management and Control of Production and Logistics, Campinas, SP, Brazil.

Booch G.: Object Oriented Design with Applications, New York: Benjamin/Cummings in David R C Hill: Object-Oriented Analysis and Simulation, 1996, Addison-Wesley Publishing Company.

Cetnarowicz E, Nawarecki E, Cetnarowicz K: Agent oriented technology of decentralized systems based on the M-agent architecture, 1997, IFAC/IFIP Conference on Management and Control of Production and Logistics, Campinas, SP, Brazil.

Coad P, Yourdon E: Object-Oriented Analysis, 1991, Yourdon Press, PTR Prentice Hall Inc, Second Edition.

Gavoronski A. A: Agent nets: a methodology for evaluation of enterprise strategies in information economy, 1998, Telektronikk 3/4, Strategy in telecommunication.

Gigh, J.P. van, System Design Modeling and Meta modeling, 1991, Plenum Press, New York.

Hill D R C: Object-Oriented Analysis and Simulation, 1996, Addison-Wesley Publishing Company.

Kjenstad D.: Coordinated supply chain scheduling. 1998, Ph.D. thesis, NTNU.

Lillehagen F.: Process Design, 1995, Presentation at MNEMOS Methodology Workshop, Aerospatiale, Toulouse, AT&T Gis.

Malhotra: Knowledge Management, Knowledge Organization and Knowledge Workers: A view from the front lines, 1998, Mail Business Newspaper of February 19.

Nawarecki, E and Marcjan, R: Attaining a compromise in multi-agent world, 1997, IFAC/IFIP Conference on Management and Control of Production and Logistics, Campinas, SP, Brazil.

Prytz K., Not Shimon Y., Rolstadås A.: Manufacturing Integration and learning by Corporate Memory Systems, Research Memorandum No 95-12, Sept. 1995, Department of Industrial Engineering, Purdue University.

Scheer, W.A. Architecture for Integrated Information Systems, Springer Verlag, Berlin in Francois B. Vernadat: Enterprise modeling and integration, Chapman & Hall, London, 1996.

Sichman J. S.: On achievable goals and feasible plans in open multi-agent systems, 1997, IFAC/IFIP Conference on Management and Control of Production and Logistics, Campinas, SP, Brazil.